AF 2857
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Before the Board of Patent Appeals and Interferences

In re Patent Application of

WILSON

Serial No. 09/680,334

Filed: October 6, 2000

Title: TEST BITSTREAM GENERATOR AND METHOD

Atty Dkt. 550-183
C# M#

TC/A.U.: 2857

Examiner: E. Desta

Date: May 21, 2004

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

☐ **Correspondence Address Indication Form Attached.**

☐	**NOTICE OF APPEAL** Applicant hereby **appeals** to the Board of Patent Appeals and Interferences from the last decision of the Examiner. ($ 330.00)	$	
☒	An appeal **BRIEF** is attached in triplicate in the pending appeal of the above-identified application ($ 330.00)	$	330.00
☐	Credit for fees paid in prior appeal without decision on merits	-$ (	)
☐	A reply brief is attached in triplicate under Rule 193(b)		**(no fee)**
☐	Petition is hereby made to extend the current due date so as to cover the filing date of this paper and attachment(s) (**$110.00/1 month**; $420.00/2 months; **$950.00/3 months**; $1480.00/4 months)	$	
		SUBTOTAL $	330.00
☐	Applicant claims "Small entity" status, enter ½ of subtotal and subtract ☐ "Small entity" statement attached.	-$(	)
		SUBTOTAL $	330.00
	Less month extension previously paid on	-$(	0.00)
		TOTAL FEE ENCLOSED $	330.00

Any future submission requiring an extension of time is hereby stated to include a petition for such time extension. The Commissioner is hereby authorized to charge any deficiency, or credit any overpayment, in the fee(s) filed, or asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our **Account No. 14-1140.** A duplicate copy of this sheet is attached.

1100 North Glebe Road, 8th Floor
Arlington, Virginia 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100
JRL:at

NIXON & VANDERHYE P.C.
By Atty: John R. Lastova, Reg. No. 33,149

Signature: _____

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

Wilson

Serial No. 09/680,334

Filed: October 6, 2000

For: TEST BITSTREAM GENERATOR AND METHOD

Atty. Ref.: 550-183

Group: 2857

Examiner: Desta, E.

Before the Board of Patent Appeals and Interferences

BRIEF FOR APPELLANT

On Appeal From Final Rejection from Group Art Unit 2857

John R. Lastova
NIXON & VANDERHYE P.C.
8th Floor, 1100 North Glebe Road
Arlington, Virginia 22201-4714
(703) 816-4025
Attorney for Appellant

TABLE OF CONTENTS

REAL PARTY IN INTEREST 1
RELATED APPEALS AND INTERFERENCES 1
STATUS OF CLAIMS 1
STATUS OF AMENDMENTS 2
SUMMARY 2
ISSUES 5
GROUPING OF CLAIMS 5
ARGUMENT 5
A. Anticipation Requires Panaro to Disclose Every Claim Feature 5
B. Panaro's Teachings 5
C. Panaro Fails to Disclose Every Claim Feature 7
CONCLUSION 12

TABLE OF AUTHORITIES

Scripps Clinic & Research Found. v. Genentec, Inc., 927 F.2d 1565 (Fed. Cir. 1991) 5

Kloster Speedsteel AB v. Crucible, Inc., 793 F.2d 1565 (Fed. Cir. 1986) 5

GIN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

WILSON — Atty. Ref.: 550-183

Serial No. 09/680,334 — TC/A.U.: 2857

Filed: October 6, 2000 — Examiner: E. Desta

For: TEST BITSTREAM GENERATOR AND METHOD

* *

May 21, 2004

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This is an Appeal from the Examiner's final rejection dated October 21, 2003.[1]

REAL PARTY IN INTEREST

The real party in interest is the assignee, ARM Limited, a corporation of the United Kingdom.

RELATED APPEALS AND INTERFERENCES

There are no other appeals related to this subject application. There are no interferences related to this subject application.

STATUS OF CLAIMS

Claims 1 and 3-13 are pending. All pending claims stand rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent 5,731,839 to Panaro.

[1] The claims on Appeal appear in the Appendix accompanying this brief.

STATUS OF AMENDMENTS

The amendment filed after final on April 23, 2004 was entered for purposes of Appeal, as indicated in the Examiner's Advisory Action dated May 10, 2004.

SUMMARY

The invention relates to generating bitstreams used to test bitstream decoders. Bitstream decoders generally decode bitstreams that have a predetermined format corresponding to a syntax that defines the format and context of bitstreams generated in accordance with that syntax. For example, the MPEG-4 standard includes syntaxes that define the structure of audio and video bitstreams generated in accordance with that standard. In this example, the syntax is defined in pseudo-C, and a number of tables are provided defining the values that most variables referenced in that syntax can take.

Prior art approaches, including Panaro's approach, *manually* generate a set of test bitstreams on a case-by-case basis. This manual approach requires the bitstream developer to understand both the bitstream standard (e.g., MPEG-4) and code coverage measurements (i.e., a measurement of how much code under test has been exercised by the test data). The developer starts with a valid bit stream and edits it so that each decoder component is tested. This editing process leads to generating a set of test bitstreams deemed complete when each decoder component has been tested. The invention provides a better technique for generating such decoder bitstreams.

The inventors determined that it was possible to generate test code from a

bitstream syntax. The test code references plural values to which variables need to be assigned to test a particular bitstream. A number of "interesting values" is determined for each test code variable. When executing the test code, each test code variable is assigned one of its interesting values. Thus, the generated a test bitstream depends on the interesting value assigned to each test code variable.

The inventive approach is particularly applicable where the syntax specification for a bitstream can be converted into compilable computer code, which can be executed by a computer to generate test bitstreams. As mentioned above, the MPEG-4 standards have syntaxes which define the format and contents of the bitstream using pseudo-C, (where "C" is a compilable computer language), and a number of tables defining the meaning of variables, which makes the inventive approach particularly effective in generating test bitstreams for MPEG-4 bitstream decoders. The test code created from the syntax is executed repeatedly until each variable is assigned each of its interesting values, and a set of test bitstreams is generated.

Each time a variable needs to be assigned a new value according to the syntax, the test code is arranged to update that value with an interesting value using a function call inserted into the pseudo-C. Conditions/loops will pass or fail under the control of additionally created variables, which take on new interesting values before each condition/loop. With such repeated calls to the syntax, eventually all variables will take on all there interesting values, at which point, a set of test bitstreams will have been generated for thoroughly testing the bitstream decoder.

Figure 3 shows a main control loop 300 which controls the number of times that the syntax computer code is executed to generate a test bitstream. Syntax code 310 is developed directly from the pseudo-C version of the MPEG-4 video syntax. Whenever a new value of a variable is required, the syntax code 310 issues a message to value assignor code 360 requesting that the variable be assigned a particular interesting value. For example, tables may be stored in storage 370 which contain for each variable a number of interesting values. The value assignor code 360 determines when all interesting values have been used for that variable. Alternatively, value assignor code 360 may invoke some predetermined code to generate interesting values without having to predefine those values within a table.

Once the syntax code 310 and derived behavior code 320 have been executed with interesting values being assigned to the variables by the value assignor code 360, a test bitstream is generated. Output bitstream manipulation code 330 performs certain operations on the output test bitstream, e.g., byte alignment operations, etc. Thereafter, the output bitstream is stored in a file 340. Once all variables have been processed using their interesting values, a set of test bitstreams is stored in that file 340, and thereafter, may be used to thoroughly test a bitstream decoder. A detailed, non-limiting, example embodiment for a video bitstream test generator design is set forth on pages 13-30.

ISSUES

The ARM appeal presents one issue.

- **Anticipation.** An anticipation rejection requires that each and every feature recited in the rejected claim be disclosed in the single prior art reference. Both independent claims 1 and 11 recite features not disclosed in Panaro. Does Panaro anticipate the appealed claims?

GROUPING OF CLAIMS

Claims 1, 4, 8, and 10-13 stand or fall together. The remaining claims are argued separately.

ARGUMENT

A. Anticipation Requires Panaro to Disclose Every Claim Feature

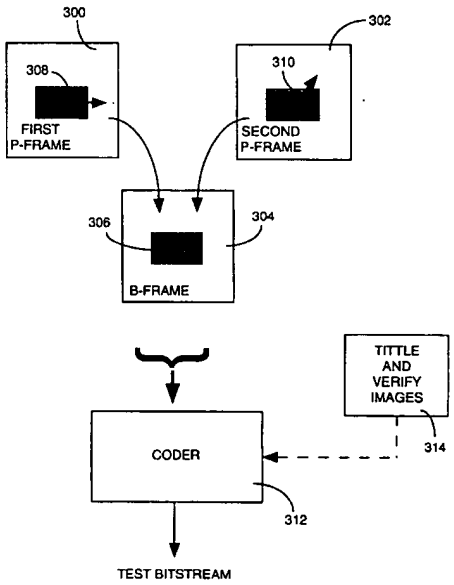
To establish that a claim is anticipated, the Examiner must point out where each and every limitation in the claim is found in a single prior art reference. *Scripps Clinic & Research Found. v. Genentec, Inc.*, 927 F.2d 1565 (Fed. Cir. 1991). Every limitation contained in the claims must be present in the reference, and if even one limitation is missing from the reference, then it does not anticipate the claim. *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565 (Fed. Cir. 1986). Panaro fails to satisfy this rigorous standard.

B. Panaro's Teachings

Panaro describes testing MPEG video decoders by inputting a known data sequence and analyzing the decoder's output data sequences to determine if the results conform to expectations. Panaro *manually* generates two "anchor" images

that have a uniform gray region located within each image called predicted images or P-frames. The gray region in each of the two anchor images is assigned a set of motion vectors having a predefined magnitude and/or direction. The magnitude and/or direction of the two sets of motion vectors is/are not the same. As shown in Figure 3 below, the first P-frame has a motion vector pointing directly to the right (East), and the second P-frame has a motion vector pointing Northeast.

FIG. 3



The two P-frames 300 and 302 are used to generate a bi-directional frame (B-frame) 304. The two P-frames and the B-frame are encoded by coder 312 using a conventional coding scheme to generate a test bitstream which is then stored RAM memory 116. The test bitstream may be used to test the effectiveness of the a decoder's capability for decoding B-frames. After inputting this test bitstream to the decoder, any errors in the decoding process will appear in the decoded image as imperfections in the uniformly gray region.

C. Panaro Fails to Disclose Every Claim Feature

The preambles of claims 1 and 11 recite "generating test bitstreams to test a bitstream decoder arranged to decode bitstreams generated in accordance with the predefined syntax." The Examiner reads "predefined syntax" on Panaro's known sequence of data, as described in column 1, lines 51-59. It is unclear how the known sequence of data identified by the Examiner is different from the test bitstream. But, in any event, Panaro fails to generate computer-executable test code that incorporates this predefined syntax.

The independent claims specifically state: "the test code being arranged when executed to generate a test bitstream dependent on values assigned to a plurality of variables, each variable having a number of interesting values." The Examiner fails to identify where Panaro describes *generating this computer-executable test code.*

In the "response" section to Applicant's comments, on page 7 of the final Office Action, the Examiner then switches what he is reading the "predefined syntax" on in Panaro and states that the predefined syntax is "basically one of the characteristics of MPEG encoding scheme." The Examiner then indicates that the encoder 312 in Figure 3 implementing an MPEG coding algorithm forms the test bitstream. The claimed plurality of variables is read onto Panaro's set of motion vectors

There is no teaching in Panaro of how the MPEG coding algorithm (the alleged test code) is generated. The most likely conclusion is that the MPEG

coding algorithm is a merely standard "off-the-shelf" MPEG coding algorithm. This conclusion is consistent with Panaro's central teachings that are concerned with choosing images from which to generate a test bitstream (column 2, lines 20-45) rather than the MPEG coding algorithm itself. Furthermore, column 5, lines 5-17 state that the MPEG coder 312 is an *conventional*, block-based, predictive video coder. Accordingly, Panaro's off-the-shelf MPEG coding algorithm is not test code as required by element (a) of claims 1 and 11—it is simply a coding algorithm. Panaro fails to teach (1) "generating" the MPEG coding algorithm or (2) that the MPEG coding algorithm is the "test code" as recited in claim element (a).

Panaro also fails to teach element (b) of the independent claims. Element (b) in claim 1 recites: "executing the test code, including the step of, for each of said variables, assigning that variable one of its interesting values, thereby generating a test bitstream dependent on the interesting value assigned to each variable."

Panaro's conventional MPEG coding algorithm must be *manually* fed with appropriate input data associated with the two P-frames. Column 4, lines 55-57 states:

> the two frames are *created by hand*, i.e., a *user selects* appropriate motion vector magnitudes and directions (offsets) that substantially generate the uniformly gray region in an image predicted from the anchor images. (emphasis added).

The manually-created first and second P-frames and the B-frame from user-selected motion vector magnitudes and directions are simply passed through the MPEG

coder 312 in order to generate a test bitstream. Executing the MPEG coding algorithm does not, as a part of its execution, assign each variable one of multiple interesting values. Instead, Panaro's the motion vector "magnitude" and "direction", identified by the Examiner as corresponding to the claimed plural variables, are selected manually by the user. These manually defined "variables" are input to—not defined by—the MPEG coding algorithm to generate a bitstream. Executing Panaro's MPEG coding algorithm does not result in the MPEG coding algorithm assigning each motion vector variable "one of its interesting values," as recited in claim element (b).

Panaro fails to disclose (1) each variable (allegedly the motion vector) having multiple interesting values, and (2) repeating execution of the test code (allegedly the MPEG coding algorithm) until the test code has been executed multiple times with each variable having been assigned each of its interesting values. These missing features are elements in both claims 1 and 11. As Panaro explains at column 2, lines 25-28, each anchor image is assigned a set of motion vectors that has "a predefined characteristic (e.g., magnitude and/or direction)." There is *only one* predefined characteristic per set of motion vectors. In other words, Panaro does not describe assigning different values to each set of motion vectors or re-executing the MPEG coding algorithm multiple times using different vector values.

In section 4 of the final action, the Examiner points to Figure 3, at column 5, lines 33-55. The Examiner refers to generating "sets of bitstream values" from the

first and second P-frames and the B-frame, suggesting that each B-frame corresponds to "interesting (predetermined) values." Figure 3 in Panaro, reproduced above, shows no repeatedly executed process in which different values are assigned to the motion vectors. While one can argue that the motion vectors correspond to a plurality of variables, there is no teaching of *using more than one value for each of those variables*. The text in column 5 simply indicates that the B-frame is repeated fifteen times. Panaro never teaches that each of these repeated B-frames is generated using different motion vector values. Instead, each B-frame simply repeated fifteen times using the *same values*. And as would be appreciated by those skilled in the art, the sequence shown at column 5, line 39, is simply *a single test bitstream*—not a set of plural bitstreams as claimed.

Element (b) in the independent claims specifies that a test bitstream is generated by executing the test code when each variable is assigned one of its interesting values, "thereby generating a test bitstream dependent on the interesting value assigned to each variable." The "wherein" clause recites that this operation in element (b) is repeated until each variable has been assigned each of its interesting values, whereby a set of test bitstreams is generated. In other words, to generate the set of test multiple bitstreams, different interesting values are assigned to each one of the variables. Certainly this is not the case in column 5, since the same B-frame is used fifteen times. The motion vector direction and magnitude values have not varied during the generation of those fifteen B-frames.

Column 5, lines 40-55, referred to by the Examiner, teach that certain motion vector values at the image edge are *excluded* from any bitstream that might be generated using Panaro's approach. As such, Panaro explicitly *teaches away* from the approach defined in the independent claims. This coupled with Panaro's failure to disclose multiple features from the independent claims, makes it clear that Panaro does not anticipate those claims.

The dependent claims also recite features not found in Panaro. Regarding to claim 3, the Examiner fails to point out where Panaro teaches generating both "supported bitstreams supported by the bitstream decoder" and "unsupported bitstreams that are valid having regard to the syntax, but not supported by the bitstream decoder," wherein "the test code is executed to generate a set of supported test bitstreams and a set of unsupported test bitstreams." Where in the text referred to by the Examiner are the claimed first and second sets of interesting values? Where are both the claimed supported and unsupported bitstreams generated?

Claim 5 recites that the bitstream decoder supports at least one of the plural variables defined by the syntax as "having any value from a set of non-overlapping continuous ranges." The Examiner fails to point out where this feature is found in Panaro. Nor does the Examiner indicate where Panaro teaches that the interesting values of that variable are the "boundary cases of each range in the set." As explained above with respect to claim 2, Panaro does not teach generating unsupported, but valid bitstreams, and Panaro further fails to teach that in this situation, the "the interesting values of said at least one variable are those values

adjacent to, but outside of each range in the set." How does Panaro's description of decoding predicted images that do not propagate errors to other images or frames (from the column 2, lines 9-19 referred to by the Examiner) teach these interesting values from non-overlapping continuous ranges?

Regarding claim 9, there is no teaching in Panaro that each internal variable used to control execution of conditional operations within the test code "may take any value within one or more ranges of values, and the interesting values for the internal variable are the boundary cases for each range." Column 2, lines 21-32 relied on by the Examiner do not mention interesting values being the boundary cases for one or more value ranges.

CONCLUSION

Panaro is fundamentally different from what is claimed in the instant application. Like the prior art schemes described in the background, Panaro is limited to manual generation of test bitstreams. In contrast, the claimed invention generates test code from a predefined syntax that when executed on a computer, assigns different interesting values to each variable to generate a set of test bitstreams. Lacking multiple features of the claims as explained above, the Board should reverse the outstanding rejection.

Respectfully submitted,

Nixon & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:at
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

APPENDIX

CLAIMS ON APPEAL

1. A method of generating test bitstreams to test a bitstream decoder arranged to decode bitstreams generated in accordance with a predefined syntax, comprising the steps of:

(a) generating test code incorporating the syntax, the test code being arranged when executed to generate a test bitstream dependent on values assigned to a plurality of variables, each variable having a number of interesting values;

(b) executing the test code, including the step of, for each of said variables, assigning that variable one of its interesting values, thereby generating a test bitstream dependent on the interesting value assigned to each variable,

wherein said step (b) is repeated until each variable has been assigned each of its interesting values, whereby a set of test bitstreams is generated.

3. A method as claimed in Claim 1, wherein each variable has a first set of interesting values for use in generating supported bitstreams supported by the bitstream decoder, and a second set of interesting values for use in generating unsupported bitstreams that are valid having regard to the syntax but not supported by the bitstream decoder, and the test code is executed to generate a set of supported test bitstreams and a set of unsupported test bitstreams.

4. A method as claimed in Claim 1, wherein at least one of said plurality of variables is defined by the syntax.

5. A method as claimed in Claim 4, wherein the bitstream decoder supports said at least one variable having any value from a set of non-overlapping continuous ranges.

6. A method as claimed in Claim 5, wherein, when generating supported bitstreams supported by the bitstream decoder, the interesting values of said at least one variable are the boundary cases of each range in the set.

7. A method as claimed in Claim 5, wherein, when generating unsupported bitstreams that are valid having regard to the syntax but not supported by the bitstream decoder, the interesting values of said at least one variable are those values adjacent to, but outside of each range in the set.

8. A method as claimed in Claim 1, wherein at least one of the variables is an internal variable used to control execution of conditional operations within the test code.

9. A method as claimed in Claim 8, wherein each internal variable may take any value within one or more ranges of values, and the interesting values for the internal variable are the boundary cases for each range.

10. A method as claimed in Claim 1, further comprising the step of generating one or more tables containing the interesting values of each variable.

11. A test bitstream generator for generating test bitstreams to test a bitstream decoder arranged to decode bitstreams generated in accordance with a predefined syntax, comprising:

a processor arranged to execute test code incorporating the syntax, the test code being arranged when executed to generate a test bitstream dependent on values assigned to a plurality of variables, each variable having a number of interesting values;

value determination means, responsive to execution of the test code, to assign to each variable one of said interesting values;

whereby a test bitstream is generated dependent on the interesting value assigned to each variable, and

wherein the processor is arranged to execute the test code respectively until each variable has been assigned each of its interesting values, whereby a set of test bitstreams is generated.

12. A computer program operable to configure a processing unit to perform a method of generating test bitstreams as claimed in Claim 1.

13. A carrier medium comprising a computer program as claimed in Claim 12.